Figure 3 shows an alternative bus structure embodiment of the fault tolerant computer system 100. The two PC buses 214 and 216 contain bridges 242, 244, 246 and 248 to PC bus systems 250, 252, 254, and 256. As with the PC buses 214 and 216, the PC buses 250, 252, 254 and 256 can be designed according to any type of bus architecture including PCI, ISA, EISA, and Microchannel. The PC buses 250, 252, 254, and 256 are connected, respectively, to a canister 258, 260, 262 and 264. The canisters 258, 260, 262, and 264 are casings for a detachable bus system and provide multiple slots for adapters. In the illustrated canister, there are four adapter slots.

Referring now to Figure 4, the present invention for monitoring and diagnosing environmental conditions may be implemented by using a network of microcontrollers 225 located on the fault tolerant computer system 100. In one embodiment some of the microcontrollers are placed on a system board or motherboard 302 while other microcontrollers are placed on a backplane 304. Furthermore, in the embodiment of Figure 3, some of the microcontrollers such as Canister controller A 324 may reside on a removable canister.

Figure 4 illustrates that the network of microcontrollers 225 is connected to one of the CPUs 200 by an ISA bus 308. The ISA 308 bus interfaces the network of microcontrollers 225 which are connected on the microcontroller bus 310 through a System Interface 312. In one embodiment of the invention, the microcontrollers communicate through an I2C serial bus, also referred to as a microcontroller bus 310. The document "The I2C Bus and How to Use It" (Philips Semiconductor, 1992) is hereby incorporated by reference. The I2C bus is a bi-directional two-wire bus and operates at a 400 kbps rate in the present embodiment. However, other bus structures and protocols could be employed in connection with this invention. In other embodiments, IEEE 1394 (Firewire), IEEE 422, IEEE 488 (GPIB), RS-185, Apple ADB, Universal Serial Bus (USB), or Controller Area Network (CAN) could be utilized as the microcontroller bus. Control on the microcontroller bus is distributed. Each microcontroller can be a sender (a master) or a receiver (a slave) and each is interconnected by this bus. A microcontroller directly controls its own resources, and indirectly controls resources of other microcontrollers on the bus.

Here are some of the features of the I2C-bus:

- Only two bus line are required: a serial data line (SDA) and a serial clock line (SCL).
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers.
- The bus is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer.
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 400 kbit/second in the fast mode.

Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the I2C bus. Each device is recognized by a unique address and can operate as either a transmitter or receiver, depending on the function of the device. Further, each device can operate from time to time as both a transmitter and a receiver. For example, a memory device connected to the I2C bus could both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see Table 1). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

TABLE 1 Definition of I2C-bus terminology

Term	Description
Transmitter	The device which sends the data to the bus
Receiver	The device which receives the data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by a master
Multi-master	More than one master can attempt to control the bus at the same time without corrupting the message. Each device at separate times may act as a master.

Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the message is not corrupted
Synchronization	Procedure to synchronize the clock signal of two or more devices

The I2C-bus is a multi-master bus. This means that more than one device capable of controlling the bus can be connected to it. As masters are usually microcontrollers, consider the case of a data transfer between two microcontrollers connected to the I2C-bus. This highlights the master-slave and receiver-transmitter relationships to be found on the I2C-bus. It should be noted that these relationships are not permanent, but only depend on the direction of data transfer at that time. The transfer of data between microcontrollers is further described in Figure 8.

The possibility of connecting more than one microcontroller to the I2C-bus means that more than one master could try to initiate a data transfer at the same time. To avoid the conflict that might ensue from such an event, an arbitration procedure has been developed. This procedure relies on the wired-AND connection of all I2C interfaces to the I2C-bus.

If two or more masters try to put information onto the bus, as long as they put the same information onto the bus, there is no problem. Each monitors the state of the SDL. If a microcontroller expects to find that the SDL is high, but finds that it is low, the microcontroller assumes it lost the arbitration and stops sending data. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line.

Generation of clock signal on the I2C-bus is always the responsibility of master devices. Each master microcontroller generates its own clock signals when transferring data on the bus.

- Only two bus line are required: a serial data line (SDA) and a serial clock line (SCL).
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers.
- The bus is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer.
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 400 kbit/second in the fast mode.

Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the I2C bus. Each device is recognized by a unique address and can operate as either a transmitter or receiver, depending on the function of the device. Further, each device can operate from time to time as both a transmitter and a receiver. For example, a memory device connected to the I2C bus could both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see Table 1). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

TABLE 1 Definition of I2C-bus terminology

Term	Description
Transmitter	The device which sends the data to the bus
Receiver	The device which receives the data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by a master
Multi-master	More than one master can attempt to control the bus at the same time without corrupting the message. Each device at separate times may act as a master.
Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the message is not corrupted

Synchronization Procedure to synchronize the clock signal of two or more devices

The I2C-bus is a multi-master bus. This means that more than one device capable of controlling the bus can be connected to it. As masters are usually microcontrollers, consider the case of a data transfer between two microcontrollers connected to the I2C-bus. This highlights the master-slave and receiver-transmitter relationships to be found on the I2C-bus. It should be noted that these relationships are not permanent, but only depend on the direction of data transfer at that time. The transfer of data between microcontrollers is further described in Figure 8.

The possibility of connecting more than one microcontroller to the I2C-bus means that more than one master could try to initiate a data transfer at the same time. To avoid the conflict that might ensue from such an event, an arbitration procedure has been developed. This procedure relies on the wired-AND connection of all I2C interfaces to the I2C-bus.

If two or more masters try to put information onto the bus, as long as they put the same information onto the bus, there is no problem. Each monitors the state of the SDL. If a microcontroller expects to find that the SDL is high, but finds that it is low, the microcontroller assumes it lost the arbitration and stops sending data. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line.

Generation of clock signal on the I2C-bus is always the responsibility of master devices. Each master microcontroller generates its own clock signals when transferring data on the bus.

In one embodiment, the command, diagnostic, monitoring and history functions of the microcontroller network 102 are accessed using a global network memory and a protocol has been defined so that applications can access system resources without intimate knowledge of the underlying network of microcontrollers. That is, any function may be queried simply by generating a network "read" request targeted at the function's known global network address. In the same fashion, a function may be exercised simply by "writing" to its global network address. Any microcontroller may initiate read/write activity by sending a message on the I2C bus to the microcontroller responsible for the function (which can be determined from the known global address of the function). The network memory model includes typing information as part of the memory addressing information.

messages to the other components of the server system 100. The processors or microcontrollers include: a System Interface 312, a CPU A controller 314, a CPU B controller 316, a System Recorder 320, a Chassis controller 318, a Canister A controller 324, a Canister B controller 326, a Canister C controller 328, a Canister D controller 330 and a Remote Interface controller 332. The System Interface controller 312, the CPU A controller 314 and the CPU B controller 316 are located on a system board 302 in the fault tolerant computer system 100. Also located on the system board are one or more central processing units (CPUs) or microprocessors 164 and the Industry Standard Architecture (ISA) bus 296 that connects to the System Interface Controller 312. The CPUs 200 may be any conventional general purpose single-chip or multi-chip microprocessor such as a Pentium7, Pentium® Pro or Pentium® II processor available from Intel Corporation, A MIPS® processor available from Silicon Graphics, Inc., a SPARC processor from Sun Microsystems, Inc., a Power PC® processor available from Motorola, or an ALPHA® processor available from Digital Equipment Corporation. In addition, the CPUs 200 may be any conventional special purpose microprocessor such as a digital signal processor or a graphics processor.

The System Recorder 320 and Chassis controller 318, along with a data string such as a random access non-volatile access memory (NVRAM) 322 that connects to the System Recorder 320, are located on a backplane 304 of the fault tolerant computer system 100. The data storage 322 may be independently powered and may retain its contents when power is unavailable. The data storage 322 is used to log system status, so that when a failure of the computer 100 occurs, maintenance personnel can access the storage 322 and search for information about what component failed. An NVRAM is used for the data